

AMENDMENTS TO THE CLAIMS:

The listing of claims will replace all prior versions, and listings of claims in the application:

LISTING OF CLAIMS:

1. (Currently amended) A method of operating a printing system for parallel processing a print job with a plurality of processing nodes into a printer ready format for printing the print job, said processing nodes communicating with a virtual disk transfer system, comprising:

splitting the print job into a plurality of job chunks, wherein the chunks are ~~selectively-sized range in size~~-from at least one page to ~~[[an]] the~~ entire size of the print job, ~~wherein pages comprising the chunks are selected in~~ accordance with predetermined selection factors for ~~enhancing-optimizing page printing~~ processing efficiency;

selectively storing the job chunks and print-ready pages in the virtual disk ~~remote~~ transfer system wherein the transfer system data comprises an intermediary storage for data transfer to selected processing nodes including a RAM and a physical disk, and where the virtual disk system is implemented by providing a shared memory interface;

assigning the job chunks to respective processing nodes for parallel processing the job chunks into the printer-ready image-format;

monitoring available space in the virtual disk transfer system including detecting a data overflow in the RAM and storing new data in the physical disk until data storage in the RAM is available; and

printing the print job.

2. (Previously presented) The method defined in claim 1, further comprising the step of preventing selected chunks from being added to the virtual disk transfer system when the monitored available space falls below a predetermined threshold representative of the overflow.

3. (Original) The method defined in claim 2, wherein the splitting step is performed by a splitter and further comprising the step of withholding chunk destinations from the splitter.

4. (Previously Presented) The method of claim 1 further including paging out the chunks of data from the virtual disk transfer system in a most-recently used order, wherein a least recently-used chunk is read soonest.

5. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes the total number of pages within the print job.

6. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes the total number of bits within the print job.

7. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes the total amount of processing required to process the job.

8. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes the amount of startup time needed to set up the job.

9. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes number of pages containing non-text images contained in the print job.

10. (New) The method of claim 1, wherein the predetermined factors used to determine the size of a job further includes number of pages containing color in the print job.

11. (New) The method of claim 1, wherein separate print queues may be defined as protected postscript and not protected postscript.

12. (New) The method of claim 1, wherein the parallel processing system records the previous predetermined selection factors and uses statistical analysis to determine the optimal size of chunks.

13. (New) The method of claim 1, wherein the parallel processing system incorporates load balancing to spread the workload out evenly among the print devices.

14. (New) The method of claim 1, wherein the parallel processing system incorporates auto discovery to evaluate the availability of hardware resources.

15. (New) The method of claim 1, wherein the splitting functionality may have the user selected status of maximum, recommended, and allocated.

16. (New) A method of operating a printing system for parallel processing a print job with a plurality of processing nodes into a printer ready format for printing the print job, said parallel processing nodes communicating with a virtual disk transfer system, comprising:

splitting the print job into a plurality of job chunks to be run separately over a plurality of parallel processors, wherein the sizes of the chunks are selected in accordance with predetermined selection factors for optimizing efficiency of the parallel processing printing;

selectively storing the job chunks and print-ready pages in the virtual disk remote transfer system wherein the transfer system data comprises an intermediary storage for data transfer to selected processing nodes including a RAM and a physical disk and where the virtual disk system is implemented by providing a shared memory interface;

assigning the job chunks to respective processing nodes for parallel processing the job chunks into the printer-ready format;

monitoring available space in the virtual disk transfer system including detecting a data overflow in the RAM and storing new data in the physical disk until data storage in the RAM is available; and

printing the print job.

17. (New) The method of claim 16, wherein the parallel processing comprises at least three processors connected to at least three separate memories.

18. (New) The method of claim 16, wherein the parallel processing is implemented in symmetric multiprocessing wherein two or more processors can connect to a single shared main memory.

19. (New) The method of claim 16, wherein the parallel processing is implemented in distributed multiprocessor

20. (New) A method of operating a printing system for parallel processing comprising the steps of:

inputting a print job;

storing the print job in a spooling system;

determining the language, size and location of the print job;

advising the supervisor to select splitter;

splitter advised of job location and chunk parameters;

assigning chunk to rasterized image processor (RIP) nodes;

splitting the job into chunks;

sending chunks to RIP node;

maintain chunk order by collector;

advising supervisor of chunk completion by splitter;

advising collector of page to chunk association;

parallel processing of chunks by RIP nodes;

advising supervisor and collector that chunk processing completed;
sending rasterized chunks to memory;
completing print job by splitter; and
advising supervisor of end-of-job by collector.